Mission Assurance: A Model-based Approach

Samuel S. Schreiner[1], Chester J. Everline[2], Joshua Bendig[3], Mark Boyles[4], Ben Kung[5], Bruce Chandler[6], Melissa Meyers[7], Jeffery A. Nunes[8]
Jet Propulsion Laboratory – California Institute of Technology, Pasadena, CA 91109

The purpose of Safety and Mission Assurance (SMA) at NASA's Jet Propulsion Laboratory is to ensure mission safety and success. As the complexity of technical systems increases, the tools, processes, and infrastructure used to facilitate SMA activities must evolve. A transition from a document-centric approach to a data-driven, model-based architecture is critical in allowing SMA efforts to grapple with complex system designs. This work describes how a sample subset of existing SMA processes can be executed in a model-based environment, while also describing ways in which a model-based architecture can further augment SMA efforts. One key aspect is supporting the complex interplay between the different SMA processes and the systems engineering effort to provide additional insight into the system from an SMA perspective. The proposed architecture for model-based mission assurance provides an avenue for assuring mission safety and success in increasingly complex systems.

Nomenclature

ASL	=	Approved Supplier List
IMCE	=	Integrated Model-Centric Engineering
JPL	=	Jet Propulsion Laboratory
MBSE	=	Model-Based Systems Engineering
NASA	=	National Aeronautics and Space Administration
NPD	=	NASA Policy Directive
PRA	=	Probabilistic Risk Assessment
SMA	=	Safety and Mission Assurance
SMA-TA	=	Safety and Mission Assurance Technical Authority

[1] Systems Engineer, Flight Systems Engineering, Jet Propulsion Laboratory, M/S 321-560.
[2] Principal Systems Engineer/PRA Point of Contact, Mission Assurance, Jet Propulsion Laboratory, M/S 122-107.
[3] Systems Engineer, System Reliability, Jet Propulsion Laboratory, M/S 122-107.
[4] Section Manager, Reliability Engineering & Mission Environmental Assurance, M/S 122-107.
[5] System Safety Engineer, System Safety, M/S 125-156.
[6] Quality Assurance Engineer, Hardware Quality Assurance, M/S 125-152.
[7] Group Supervisor, System Reliability, M/S 122-107
[8] Principal Engineer, Reliability Engineering, M/S 156-206.

I. Introduction

At NASA's Jet Propulsion Laboratory, Safety and Mission Assurance (SMA) plays the critical role of ensuring missions are executed successfully. As the complexity of space systems continues to increase, the tools, processes, and infrastructure used to facilitate SMA must evolve. A transition from document-centric approaches to digital model-based methodologies is critical in allowing SMA functions to provide adequate assurance for complex systems. The purpose of SMA essentially boils down to understanding and managing risk across various aspects of a project. This work presents progress towards architecting SMA into the Integrated Model-Centric Engineering (IMCE) environment at JPL. We describe the characteristics of the architecture required to support SMA disciplines, rather than the entire IMCE architecture.

Naturally, the SMA and Systems Engineering efforts are complementary and by no means are independent; assessing and managing risks are crucial functions for both. In the IMCE environment, the SMA community will provide processes that can be used to assess and manage risks, from the level of individual system components up to the system itself. Additionally, the SMA community will furnish processes for assuring the adequacy of system models, as well as verifying and validating the infrastructure used to build system models.

This work specifically addresses incorporating the following SMA processes into JPL's IMCE environment: Supplier Selection, Hazard Analysis, Probabilistic Risk Assessment (PRA), Requirement Waivers, and Risk Management. Our intent is not to redefine how these SMA processes should be conducted, rather we describe how they can benefit from being integrated into a model-centric environment. Each section describes the data in the system model, such as design elements and requirements, needed to support a given SMA process as well as the data produced by that process that can be stored in the system model to provide additional benefits. Often, the data produced by one SMA process are used as inputs to another. A model-based environment can streamline this data flow while also reducing the potential for ambiguity and errors.

A model-based approach to mission assurance must go beyond a simple conversion of existing SMA functions to model-centric approaches; the proposed model-based architecture for mission assurance must also address both risks and opportunities presented by utilizing a model-based environment. We identify areas in which a model-based environment can augment traditional SMA processes to provide additional insight into the system from an SMA perspective. One area is automated system model validation, which can enable a better understanding of the health, quality, and validity of the data in the system model. Furthermore, the proposed model-based architecture captures the dependencies between model elements that are utilized and created by each SMA process. This complex web of dependencies provides a powerful mechanism for capturing the interrelatedness that is inherent in most modern systems. The proposed architecture of model-based mission assurance provides an avenue for assuring mission safety and success in increasingly complex systems.

Three text styles are used through this paper for clarity. When describing software tools within the proposed model-based architecture, the **Tool Name** is bolded and capitalized. When describing data products in the proposed model-based architecture, the `data product` name is Rockwell font. The names of specific Databases are capitalized.

II. General Aspects of Model-based Mission Assurance

A. Data Consistency and Integration

As a first step, increased integration and accessibility of data sources can promote consistency between SMA and Systems Engineering analyses. For example, the consistency between various parts-related SMA analyses can be increased simply by ensuring that each analysis pulls data from the same source. The dependencies between SMA analyses and the data they utilize can also be leveraged to understand the impact of changes to those data sources. For instance, when a new version of the system design is released, the owners of affected analyses will be automatically notified so they can update their analyses if necessary. Section G explores the nuances of supporting consistency in system design information across multiple tools.

The benefits of this integration, of course, go beyond the system design. For example, the risk of engineers using outdated requirements is decreased by ensuring that the requirement set is readily accessible from a single database. Often, SMA functions require information from the systems engineering effort to conduct its analyses. By the time the information is released, transferred to SMA, and analyzed, the system design and requirements may have significantly evolved, creating a lag between the SMA and systems engineering efforts. The proposed model-based architecture significantly reduces this lag by enabling SMA to conduct their analyses directly on the system model being used by the systems engineering effort.

B. Injecting Mission Assurance earlier in the Project Lifecycle

In a model-based environment, many SMA analyses can be incorporated into the design process much earlier because the data is more readily accessible. This facilitates the consideration of SMA perspectives during early design efforts, reducing costs caused by adjusting the design in later phases of the project. For instance, conducting preliminary reliability analyses for conceptual designs can inject a reliability perspective earlier into design efforts to avoid costly design changes further down the road. Section E also describes how Hazard Analyses can be conducted earlier in the system design cycle using a model-based environment.

C. Assuring the System Model

In a traditional systems engineering effort, the quality of the requirements and system design is often assessed by expert analysis and review. Although model-based tools cannot replace the value of experienced engineering review, there are some aspects of this assurance effort that can be offloaded to software tools in a model-based environment. Due to the fact that data is rigorously captured in the system model, a set of automatic model validation audits can be created and routinely run on the system model to assess its health and the quality of its data. This audit functionality is allocated to the **Model Audit Tool** in the model-based infrastructure. This section identifies an example set of audits that could be automatically conducted by the **Model Audit Tool** to augment human analysis and review. It is important to understand that these audits will not replace traditional assurance activities, they merely offload some of the more tedious checks to software within the model-based environment.

The first class of audit rules concern ontologies, which are a set of concepts and categories in a domain that shows their properties and the relations between them. An ontology is essentially an agreement on how data should be embedded in the system model such that it can be unambiguously interpreted. Ontological audits compare data in the system model against the formal ontological constructs to identify where the data in the system model does not conform. These audits provide a health check of the model data - they ensure that the manner in which data is stored in the model conforms to standards such that it can be unambiguously interpreted by other tools. Without ontological audits there is no assurance that the data in the model is encoded correctly and can be safely interpreted and used.

The second class of audit rules assess the completeness *(does the data exist?*) and quality (*is the data good?*) of the data in the model. Naturally, the appropriate set of the completeness and quality checks will depend on the model-based approach used by a given project; modeling experience at JPL has shown that each project will tailor their modeling approach to meet specific needs. However, there will still exist a set of audit rules developed and negotiated at the institutional level that represent general criteria to which all system models should conform. In addition to the general institutional audit rules, there will be an additional set of audit rules that each project will for their own assessment purposes.

Table 1 provides a set of examples of potential audits across a variety of domains. In practice, a comprehensive multi-domain effort will need to be conducted to formally negotiate these rules. The purpose of including an example set in this work is to describe the general functionality required of the **Model Audit Tool**.

Table 1. Examples of audits that could be conducted for a variety of domains within the system model. This list is merely an example intended to describe the general functionality that must be provided by the Model Audit Tool, the official set of audit rules must be negotiated between projects and the institution.

Audit Category	Assessing Model Completeness *(does the data exist?)*	Assessing Model Quality *(is the data good?)*
Requirements	• *Requirement has a point-of-contact, key/driving assessment, text, connection to design element, etc.*	• *Requirement text has been marked as reviewed and approved (at a change control board, gate reviews, etc.)* • *Requirement parent-child graph is acyclic* • *No requirement is duplicated by another requirement (text check)* • *Full set of requirements are satisfiable (via constraint solving, if requirements are formally captured as constraints)*
Design Elements	• *Design element has implementer/supplier* • *Design element exists within hierarchy (has at least one composition parent)*	• *Design element has only 1 physical composition parent (may have other aggregation parents)* • *Design element hierarchy is acyclic*
Properties	• *All properties are assigned to some element in the system model (design element,*	• *Property value (mass, reliability, etc.) has a supporting analysis* • *Property value has links to elements that depend upon, and affect, its value*

	environment, etc.)	
Interfaces/ Connections	• *Ports, Connectors, and Flows have assigned types*	• *The type (and direction, where applicable) of flow elements is consistent with the associated ports and connectors*
Mission Assurance	• *Each section below will describe relevant audits for the given SMA process*	

The examples in Table 1 provide many examples of the delineation between the roles of software vs. engineers in conducting model audits. For instance, the **Model Audit Tool** can check if a given **property value** is linked to a supporting **analysis** in the system model, but it cannot evaluate whether that **analysis** actually supports the **value** in the model. The engineer must perform that supporting analysis and document it, but once it is stored in the system model the **Model Audit Tool** can then do broad evaluations of all **property values** to assess how many have supporting justification.

In addition to conducting model audits, the **Model Audit Tool** needs to support storing and accessing historical **model audit results** as shown in Figure 1. Audit results must be stored as more than a simple PDF, the **model audit results** must be tied to the relevant model elements and audit rules. This allows a user to view the audit history of a particular model element to better understand how it has evolved over time. For instance, if a requirement fails a given audit, the user may wish to investigate the historical **audit results** for that requirement to understand if the failure is due to a recent change or if it has existed for a longer period of time. Storing **audit results** in the model also allows users to investigate historical trends over time. For instance, a subsystem lead may wish to view the audit results for only their requirement set over a given development period to evaluate how much progress has been made using an objective metric. By storing **audit results** in the model and making them accessible, the model-based environment provides a powerful tool for assuring the quality of the data over the project lifecycle.

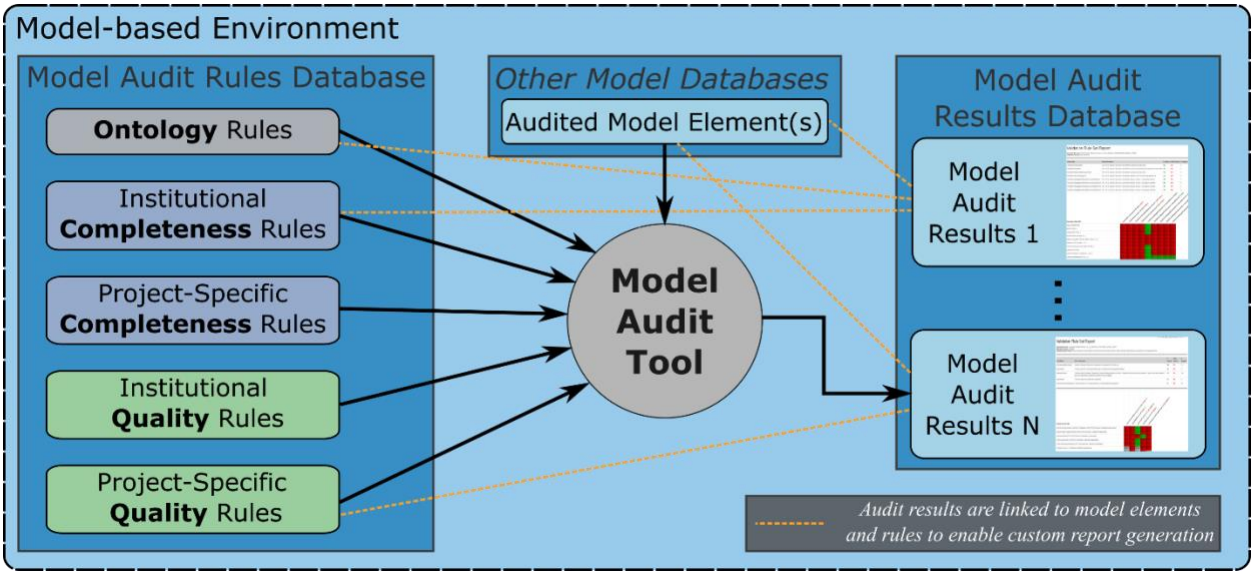

Figure 1. The Model Audit Tool executes model audit rules on a set of model elements and then stores model audit results in the model. Model audit results are linked to the relevant rules and model elements.

D. Assuring the Modeling Infrastructure

At JPL the modeling infrastructure is being developed in a manner analogous to the approach applied to flight projects. The key difference is that the product of the modeling project is software instead of a flight system. With respect to flight projects, the responsibility for assurance ultimately resides with the SMA Technical Authority (SMA-TA). Section 3.4.1.1.5 in NASA NPD 000.0B stipulates that decisions related to technical and operational matters involving safety and mission success risk require formal concurrence by the cognizant Technical Authorities (Engineering, Safety and Mission Assurance, and Health and Medical). This concurrence is based on the technical merits of the case and includes agreement that the risk is acceptable. For the IMCE modeling infrastructure development project at JPL a similar, though less formal, role has been created. This role brings both assurance and

software development perspectives, so modeling infrastructure assurance is being accomplished by adapting the standard SMA-TA functions to a project whose product is software instead of a flight system.

Even after the initial deployment of the modeling infrastructure, it is anticipated that routine updates will be made and features added. Any modification to the modeling infrastructure must also be evaluated to assure its quality. A process for assessing and evaluating the modeling infrastructure, including unit tests and usability testing, must be developed and followed to ensure that the final software product will safely interact with data in the model while meeting user expectations.

III. Design-related Assurance

This section describes the subset of SMA activities focus on managing risk within the system design.

E. Hazard Modeling

Hazard analysis is the process of identifying hazards in the system design and methods for controlling them to mitigate the risk posed during procurement, integration, testing and operations. Each hazard has an associated severity level, which is a qualitative categorization of hazard's severity based on potential consequences and probabilities (Catastrophic, critical, marginal, or negligible)[1]. Hazardous design elements are subject to an additional set of requirements that constrain the way they are handled and the way they are incorporated into the system design. For instance, severe hazards (e.g. – a high-powered laser) require a certain number of inhibits (independent and verifiable mechanical or electrical devices that prevent a hazardous event from occurring) to prevent unintentional operation.

The proposed architecture enables hazard identification early in the design process. As illustrated in Figure 2, when new **design elements** are added to the system model a piece of software called the **Hazard Analysis Tool** evaluates whether or not the given element is a hazard by comparing it to the Common Hazard Database, a database of common hazards maintained at the institutional level. If the new design element matches something from the Common Hazard Database, it is flagged as a potential hazard and additional information is gathered from the design engineer. Based on the engineer's answers, it may be possible to definitively flag the **design element** as a hazard or further analysis may be necessary. In either case, the information is stored in the model in the form of a **hazard data object**, linked to the **design element**, and a notification is sent to both the design engineer and the system safety engineer (who oversees hazards for a given project).

For definitive hazards, the engineer uses the **Hazard Analysis Tool** to determine the severity level, potential consequences, and the number of required inhibits and stores that information in the **hazard data object**. The design engineer or system safety engineer may then use the **Hazard Analysis Tool** to connect the **hazard data object** to other **design elements** that serve as inhibits to the identified hazard. For potential hazards, further analysis may be necessary to determine whether or not it is a hazard. The system safety engineer may use the **Hazard Analysis Tool** to work through the list of potential hazards in the system model, assigning them to the domain experts who are needed to conduct the additional analysis.

The **Hazard Analysis Tool** must enable users to interact with data contained in the **hazard data objects** and create reports. The model-based infrastructure must support dynamic hazard report generation on-the-fly, and these reports can also provide a convenient vehicle for modifying data relating to hazards. For instance, each hazard reduction control (e.g. "attach warning labels", "implement arming plug for laser power on S/C skin") must be verified on the as-built system, and the engineer that verifies those controls can use an interactive hazard report (in say, a web browser) as a method for storing the verification information.

The **Hazard Analysis Tool** also contains a "white list" of **design elements** that are assuredly not hazards. As new **design elements** are added that match entries on the "white list", they are labelled explicitly as non-hazards and the system safety engineer is notified for concurrence. In the model-based infrastructure, each **design element** should be labeled as either a hazard, non-hazard, or potential hazard requiring further analysis. This provides a mechanism for ensuring that each **design element** has been evaluated to determine whether or not it is a hazard, and system model audits (see Section C) can be designed to evaluate how many **design elements** are missing a hazard classification to avoid the risk of un-evaluated design elements.

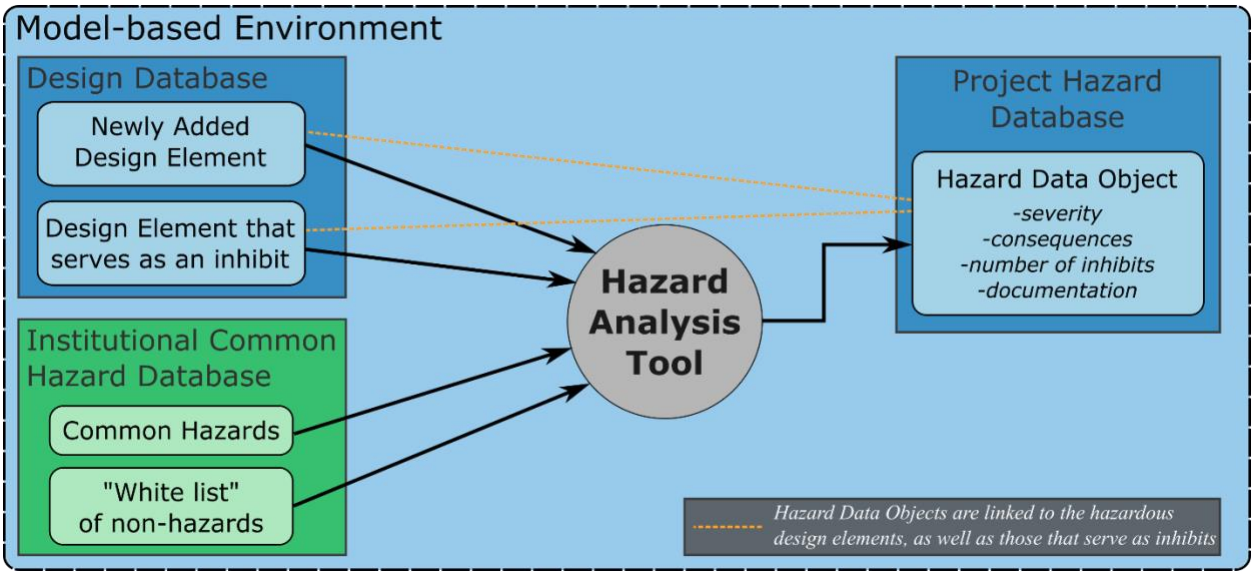

Figure 2. The Hazard Analysis Tool evaluates newly added design elements to determine if they are hazards by evaluating it against a list of common hazards and a "white-list" of non-hazards from the Institutional Common Hazard Database. If the design element is determined to be a hazard, a hazard data object is created and linked to the hazardous design element, as well as any design elements that may serve as inhibits.

F. Supplier Selection Modeling

Selecting suppliers for elements of the system design, especially quality-critical items, is another avenue where risk must be managed. To reduce the overhead and manual communication required for procurement, supplier databases must interface with the system modeling environment. A piece of software called the **Supplier Selection Tool** can provide supporting information from the system model for a given **design element** to aid in the supplier selection process, including the name, description, performance requirements, cognizant engineer, etc. Work is underway at JPL to develop an Institutional Single Supplier Database that will contain information for potential suppliers, including past performance data, ratings, points of contact, etc. For quality-critical items, suppliers must be selected from the Approved Suppliers List (ASL), which contains a subset of the suppliers in the Single Supplier Database. As illustrated in Figure 3, the **Supplier Selection Tool** queries the system model and the Single Supplier Database to compile information on the **design element** and all potential **suppliers**, and then this information is utilized in the nominal supplier selection process at JPL. In the context of procurement support, the analysis being performed is the evaluation of candidate suppliers in order to select the best vendor.

After a supplier is selected, the **Supplier Selection Tool** creates a link between the **design element** and the selected **supplier** from the Single Supplier Database. A mapping between **suppliers** and **design elements** can help identify which areas of the design, if any, might be impacted by supplier delays or manufacturing defects. Any risks (e.g. relating to technical performance, cost, schedule, or safety) identified during the supplier selection process will, as part of the system model, be readily integrated into the overall project risk profile for appropriate management. One potential supplier-related model audit (see Section C) could check if elements of the system design have an identified supplier, providing an avenue for assessing the risk of design elements that lack a supplier. A more complex audit could evaluate if any **design elements** on the critical path have a supplier with a history of schedule slips or poor ratings.

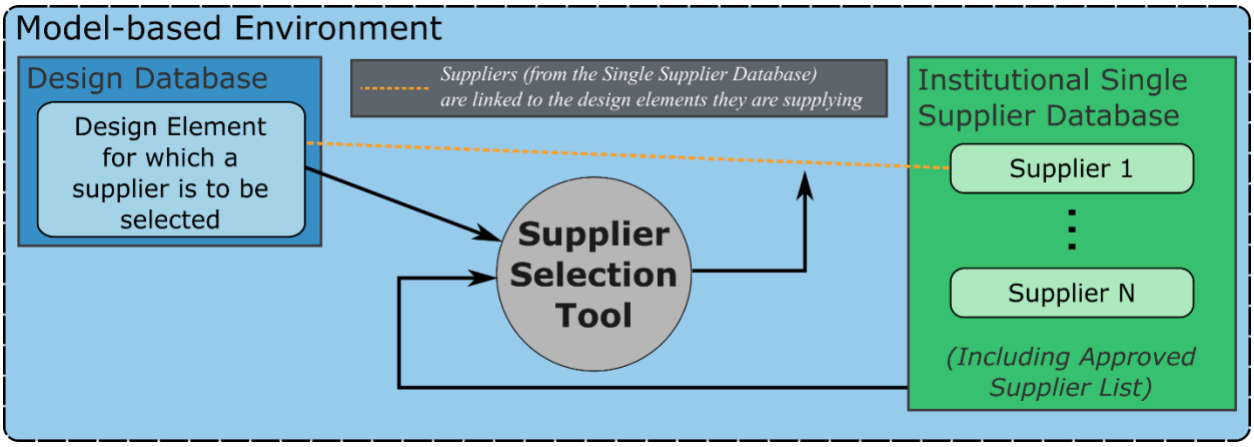

Figure 3. The Supplier Selection Tool is used to support the procurement process. The Supplier Selection Tool pulls data from an Institutional Single Supplier Database to identify potential suppliers, and once a supplier is selected, the Tool creates a connection between the selected supplier and the design element being supplied.

G. Consistent Design Information across Multiple Tools

Maintaining a consistent description of the system design across multiple tools can present a potential risk in the form of different analyses using outdated versions of the design. It is important to note that this risk also exists in non-model-based approaches, but methods of managing it are difficult with limited effectivity. The proposed model-based architecture is intended to facilitate seamless updates of the system design information at regular intervals.

To the extent possible, tooling external to the modeling environment should pull design info directly from the model (to maintain a single source of truth), but in reality there will often be tools that cannot read directly from the source-of-truth database. For such tools, the data in the IMCE environment represents the single-source of authority for the system design and the external tools will periodically update their separate copies of the system design from the IMCE source-of-authority. The functionality of exporting the system design for use by external tools is allocated to the **System Design Exporter**.

Several SMA tools were surveyed to understand the design element attributes and relationships that each tool required. For instance, one SMA tool is used to evaluate the environmental tolerance of certain **design elements** against their expected **environments**. To properly evaluate **design elements** in this tool, the tool must have access to the name, position within the design hierarchy, environmental tolerance, and expected environment for each **design element**. Another tool exclusively analyzes electrical components, so it must have access to the type attribute of a **design element** in order to determine if it is electrical. Table 2 shows a summary of all of the **design element** attributes and relationships required to support the existing SMA tools used at JPL.

Table 2. The attributes and relationships that need to be exported for each design element for use by various SMA tools external to the modeling environment.

Parameter	Description
Equipment Name	The full name of the equipment used by the project.
Acronym	The abbreviation typically consisting of several letters.
Flight System Element Identifier	This is a short acronym used by the project that indicates identifies the item in relation to the flight system, such as how an item may be used in the system, and may include an indication of redundancy (e.g. A or B)
Flight System Reference Designator	This identification is commonly use in the Flight System Function list and is used to build the system harnesses.
Logical Parent	This relationship pertains to an analytical property and is used to construct a logical hierarchy.
Physical Parent	This relationship pertains mechanical integration and indicates containment. It is used to construct a physical hierarchy.
Equipment Type	This attribute indicates characteristics about the hardware pertaining to what engineering discipline is typically evolved in its development, such as: electronics, mechanical, etc.
Hardware Types	These represent the flight article attributes that are important for the planned uses.

	Predefined list and definitions are in JPL Rules DocID 78027.
Serial Number	This is a unique identifier assigned at the time of manufacturing.
Cognizant Engineer (CogE)	Cognizant Engineer is used for notification and workflow.

H. Capturing the Reliability of Design Elements

Reliability is a key aspect of a system design, but reliability assessments are often conducted late in the design process. This makes it difficult for the design to respond to any reliability issues that are uncovered. A key barrier in this process is the availability and accessibility of reliability data for a given design. Work is underway at JPL to develop an Institutional Reliability Database that contains reliability-related information from past missions and tests. This database will be curated by a team of reliability experts and contains its own user interface for making queries.

Although it may seem appealing to develop a route by which project engineers can directly access this database and select reliability values for their design, this approach is not permitted in the proposed architecture. Selecting appropriate reliability values requires an in-depth understanding of both the existing design and the context of the existing values in the institutional database. The approach supported in this architecture involves two parties with distinct roles: 1) The project engineer, who uses the system model to gather the supporting (reliability-relevant) data for a given **design element**, and 2) the reliability engineer, who receives the reliability data request from the project engineer, including the supporting information, and then uses that information to query the Institutional Reliability Database. The reliability engineer then follows the process outlined in Appendix A to elicit expert opinions and perform analysis to select the applicable reliability data for the given request. The selected reliability data, including the supporting justification, are sent back to the project engineer, who stores the data and its justification in the system model as a **reliability value object** with a link to the **design element** being characterized. Appendix A furnishes an overview of the reliability selection process, and a detailed process for assessing hardware heritage is the subject of Ref. 2. Table 3 describes the minimal set of information that should be included in a reliability data request for a given **design element** in order to enable the reliability engineer to identify applicable reliability data from the Institutional Reliability Database.

Table 3 The set of data extracted from the system model and other databases when requesting reliability data for a given design element. See Appendix A for more details.

Data	Example	Rationale
Component name/type	RAD750, Latch Valve, Stellar Reference Unit (SRU)	Provides an easy way for the engineer to identify the component and its type.
Component description	Flight computer, composed within the Spacecraft and responsible for executing commands and processing data	A detailed description is required to help the reliability engineer understand the component for which reliability information is being requested. Ideally this data will be provided via a link to the system model with further information.
Vendor/Supplier	BAE, Honeywell, Galileo Avionica	The reliability of a given design element will likely depend on the vendor
Vendor/Supplier manufacturing data	Attributes include: past performance data, supplier rating, etc.	Information relating to the vendor qualification and testing programs, design processes, and quality program can appreciably influence hardware reliability (see Appendix A). This data allows the reliability engineer to judge whether recent changes to a vendor's processes (e.g., their quality assurance program) could alter the reliability of their products relative to the reliability those products have demonstrated in the past.
Relevant GIDEP alerts	N/A	If a GIDEP alert has been issued for the design element, its reliability may be affected.
Model number	DE1304257681	The model number permits the reliability engineer to use model-specific data (when available) and to assess whether there are certain attributes unique to that specific model which can impact the

		reliability values used in the PRA.
Operational exposure units	Operating hours, cumulative revolutions, discharge/charge cycles, on/off demands	The reliability of an element depends on the operational usage (operating hours, cumulative revolutions, discharge/charge cycles, on/off demands, etc.). For example, if the operating exposure for a mechanism is X cycles for the new mission, but heritage data is available only for exposures up to X/3, unexperienced aging and wear out phenomena could potentially represent a risk to the component's performance during the new mission.
Operating environment	Temperature, Total Ionizing Dose (TID)	The environment can affect the reliability of a design element. For example, data from a component operated in Low Earth Orbit may not be applicable for a mission to Jupiter.
Other reliability analyses	WCA, parts derating	The results from other reliability analyses may inform the selected reliability value.
Heritage	This component was flown on the Galileo mission	Most JPL projects are required to perform a heritage review in order to apprise stakeholders about potential risks associated with flying technology for the first time. This heritage review is an ideal source of information the reliability engineer can use as a guide to applicable flight and test data from these previous missions.

IV. Assurance between Requirements and Design

In additional to managing risk in the system design, it is imperative to manage risk between the project needs, as expressed in the requirements, and the capabilities of the system design.

I. Probabilistic Risk Assessment

A Probabilistic Risk Assessment (PRA) aims to identify and assess potential risks to system technical performance requirements for the purpose of furnishing risk insights into project decisions. This assessment provides an integral view into the reliability of complex systems while also identifying risk drivers. The assessment of system reliability from a PRA perspective requires knowledge of the system design, behavior, and intended operation. In a traditional engineering endeavor, a labor-intensive effort must be made to gather this data from a wide range of personnel on the project and, by the time all of the data is gathered, the system design may have changed. Directly utilizing data in the system model to support this analysis can increase consistency between the PRA and the system design. These benefits allow the PRA to provide reliability insight into the system design process earlier and more often.

A model-based approach for conducting PRAs has been previously developed and tooling developed[3]. This PRA tooling will be integrated into the proposed model-based architecture and carried out by the **PRA Tool**. The methodology starts from selected `requirement(s)` and identifies `behavior` in the system model that may constitute a threat to the selected `requirement(s)`. This first layer of `behavior` is often at a high, abstract level (e.g. the spacecraft being unable to execute a maneuver) so the **PRA Tool** traverses through `causal dependencies` in the system model to locate the underlying `behavior` that poses a threat to the selected `requirement(s)`.

The **PRA Tool** stores the results from a PRA in the system model as a `PRA data object` and links it to the relevant `requirements`, `design elements`, `behavior models`, `reliability value objects` (described in Section H) and other model elements. These links are crucial for supporting other analyses. For instance, when model elements are updated (e.g. a `reliability value` is changed), the link between that `reliability value` and the `PRA data object` can be used to notify the PRA owner that the PRA may need to be re-evaluated.

There are a number of system model audits (see Section C) that could be used to validate the model elements used by the **PRA Tool**. One rule could check if all of the `reliability values` used in the PRA have supporting justification (as described in Section H), while another rule could check if the `causal dependencies` are documented. Although these audit rules alone cannot ensure the quality of the PRA, they can still provide insight into the quality of the information in the system model used to support the PRA.

J. Waivers

Tracking areas where the design is not expected to comply with requirements is another way to manage risk. Waivers are filed to formally identify aspects of the design that will not be in compliance with requirements. In this model-based architecture, the **Waiver Tool** is used to create and manage waivers for both project requirements and

institutional requirements (e.g. JPL Design Principles). The **Waiver Tool** enables users to select a **requirement**, file a waiver for specific **design elements** and tag project **personnel** as interested parties to be notified. Much of the supporting information for a waiver can be automatically queried from the system model. For instance, the **requirement** owner and the cognizant engineer of the **design element** can be automatically suggested as interested parties. Furthermore, details pertaining to the **requirement** (text, ID, etc.) and **design element** (description) can also be automatically added. The waiver is stored as a **waiver data object** and linked to the **requirement**, **design element**, and selected **personnel** from the Project Personnel Database.

The explicit links between **waiver data objects**, **requirement**, and **design element(s)** provides a number of benefits. For instance, when a **waiver data object** is created or changed, the relevant personnel can be notified via email with a link to view the waiver in the system model. Such links also enable software tools to notify relevant personnel when changes to the requirements or design may affect a waiver. The links also enable a better understanding of where the risk lies between the requirements and system design. For instance, if a large number of waivers are filed within a particular subset of the design or for a particular subset of the requirements, the concentrated mismatch between the needs of the project (captured in the requirements) and the capability of the design may pose a threat to the overall objectives of the mission.

There are a number of system model audits that can provide insight into the quality of the waivers in the system model. Rules could be designed to check if **waivers data objects** have an assigned point of contact and explicit links to at least one **requirement** and **design element**. Waivers that are missing required connections or information pose a risk of missing pertinent notifications when relevant model elements are modified.

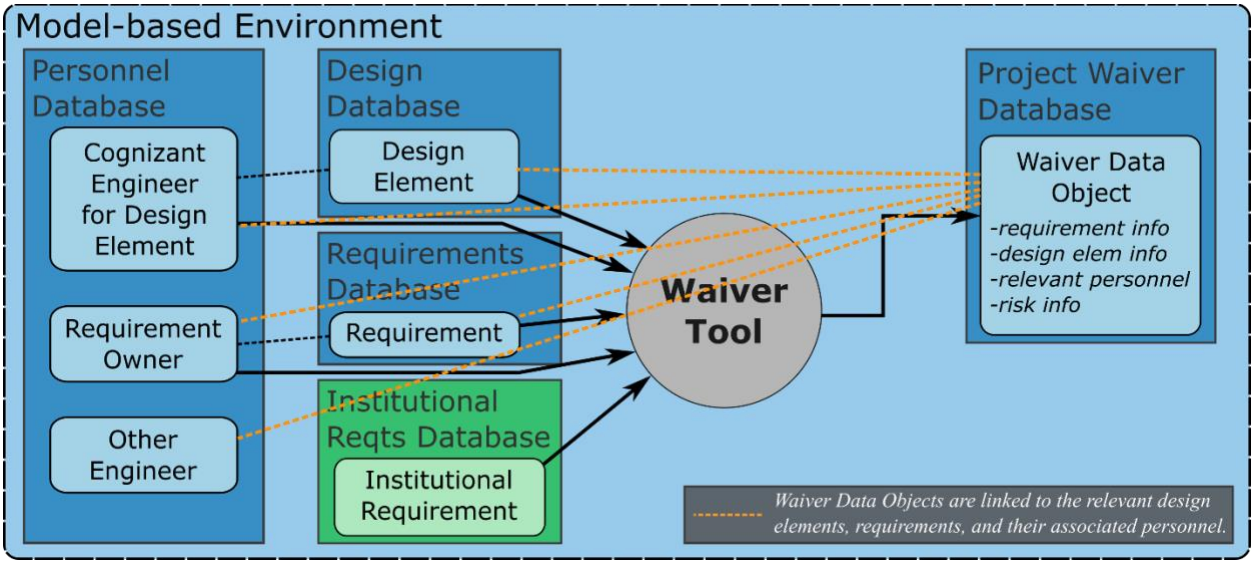

Figure 4. The Waiver Tool allows users to select requirements and design elements to identify areas of non-compliance, which are formally stored as waiver data objects. Links to personnel allows for automated notification.

V. Managing Risk across the Entire Project

The previous sections have described how to integrate existing SMA processes into a model-based environment. Although each of these processes helps manage risk for a certain aspect of the project, projects also run an overarching Risk Management task in order to better understand cross-cutting, high-impact risks to the project. Appropriately managing risks is key to enhancing the probability of mission safety and success, but when the dependencies between risks and their contributors are not captured formally, the effort of manually managing those dependencies quickly becomes overly burdensome.

Risk Management can greatly benefit from a model-based approach that formally captures these dependencies. Figure 5 illustrates how risks are captured by the **Risk Tool** in the project's Risk List Database in the form of **risk data objects**. When a **risk data object** is added to the Risk List Database, the **Risk Tool** can link it to **risk contributors** in the system model (orange lines in Figure 5). Risk contributors can be any element in the system model that contributes to the risk (e.g. system design, requirements, cost, schedule, etc.). For example, consider a mission that is planning on using a lithium-ion battery as an energy storage device. One potential risk may be related

to the handling of the out-gassing and volatility of the battery as a safety hazard (e.g. the possibility of fire and/or explosion). This risk would be related to the lithium-ion battery as a threat to safety requirements that apply to hazardous items, as illustrated by "Risk 1" in Figure 5. Risks are often related to other SMA analyses, yielding a more complex dependency. For example, a risk may be based on the results of a PRA that indicate there is a high probability that a **requirement** will not be met by a particular **design element** (this is illustrated by "Risk 2" in Figure 5). If the **reliability data** of a **design element** used in the PRA analysis (modeled as a **PRA data object**) changes, the owners of the PRA and the risk (from the Project Personnel Database) can be notified with a record of what changed and a request to determine if the PRA needs to be re-evaluated, which may impact mission risk. Figure 5 illustrates an example of how **risk data objects** may be connected to a variety of elements within the model-based architecture.

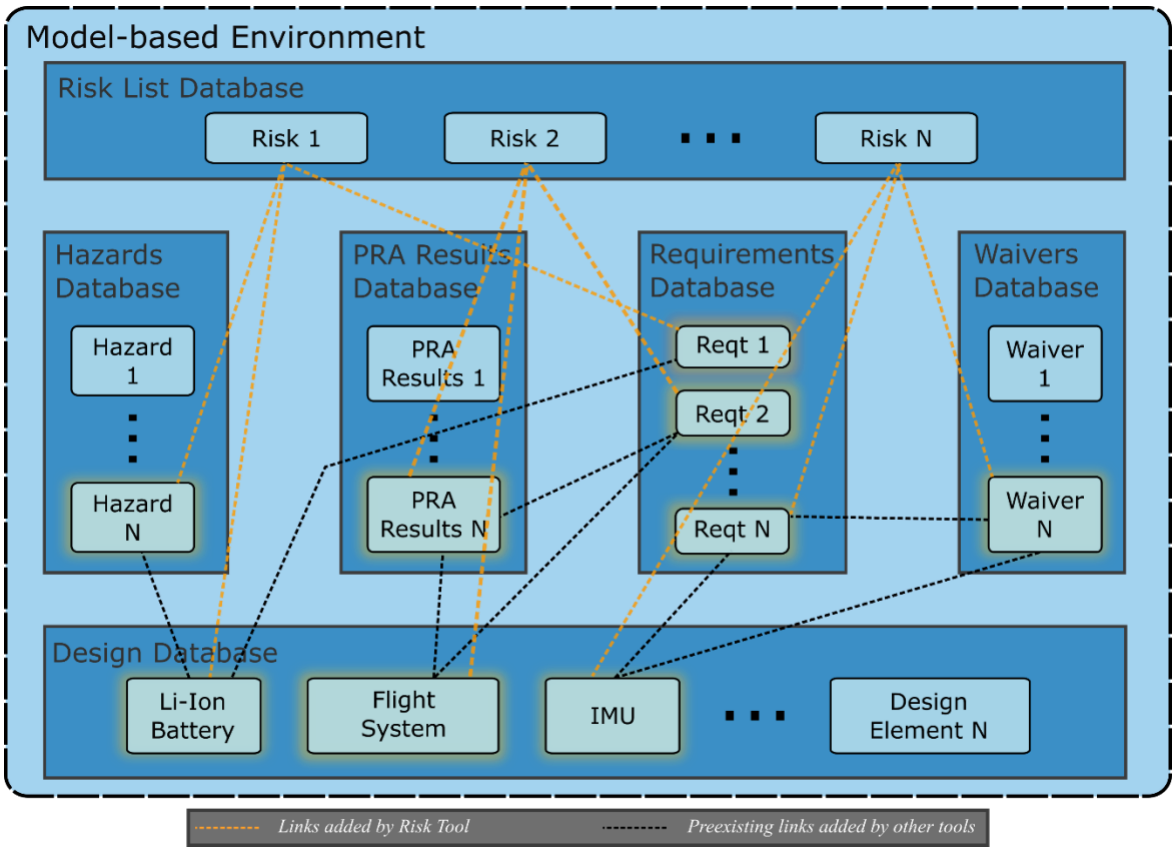

Figure 5. A graphic illustrating the complex dependencies between risks in the risk list database and risk contributors in the system model, such as requirements, waivers, hazards, and design elements. Orange lines represent links added by the Risk Tool, while black lines represent preexisting connections that were created by other tools and leveraged by the Risk Tool.

VI. Future Work

This work by no means represents the full model-based architecture needed to support SMA on a mission. Additional areas that need to be investigated include, but are not limited to, Worst-Case Analyses (WCA); Hardware and Software Requirements Certification Reviews (HRCRs/SRCRs); Failure Modes, Effects and Criticality Analysis (FMECA); GIDEP Alerts, and Fault Tree Analysis (FTA). Furthermore, the area of system model validation as a means for mission assurance remains to be fully explored. As JPL's IMCE environment is further developed, the integration of SMA into that environment will certainly evolve in parallel.

VII. Conclusion

This work describes how a subset of the SMA processes can be integrated into a model-based environment. Section II describes general aspects of mission assurance in a model-based architecture, including consistency of

information (Section A), injecting SMA analyses earlier in the design process (Section B), assuring the system model (Section C) and assuring the modeling infrastructure (Section D). Section III describes how various design-related assurance processes can be conducted using a model-based environment, including hazard analysis (Section E), supplier selection (Section F), maintaining consistent design information across tools (Section G), and capturing the reliability of design elements (Section H). Section IV discusses how two requirements-related SMA processes, Probabilistic Risk Assessment (Section I) and filing waivers (Section J), can be integrated into a model-based architecture. Section V describes how the process of risk management can greatly benefit from connections to risk contributors in the model-based environment.

Although each process benefits from the model-based environment on its own, the true value comes from capturing all of the SMA information in a single modeling environment alongside the other systems engineering information (requirements, design, behavior, etc.). The complex web of dependencies between SMA analyses, their artifacts, and other model elements can be leveraged to generate enormous insight into complex systems throughout the design process. Many of the sections above describe different dependency checks that can determine when changes to model data may affect other areas of the system design and specifications (e.g. notifying a PRA owner when the reliability value of a design element used in the PRA analysis changes). Although each dependency check may seem trivial on its own, the model-based architecture described in this work offers a mechanism for effectively managing the network of these dependencies to provide insight into the complex interrelatedness of modern systems. Although certainly not complete, our architecture for model-based mission assurance provides an outline for the type of data and functionality that must be supported by a data-driven, model-based environment to help manage the increasing complexity of space systems.

Appendix A: Overview of Reliability Selection Process

A.1 Relevant attributes for determining provenance of reliability data

Determining the applicability of an input probability for a PRA involves satisfying four criteria relating to whether:

1. the calculated failure probability corresponds to the definition of failure in the context of functional performance requirements;
2. applicable operating and environmental conditions are modeled;
3. no potentially significant risk contributor is omitted unless explicitly documented in the PRA; and
4. the probability quantification is either insensitive to the choice of input data and models (especially probabilistic models) or, if it is sensitive, there is a rationale given for the choice of data and models.

If there is sufficient provenance to conclude that all four criteria are satisfied, the input probability is considered applicable. Otherwise, the applicability is problematic.

The most direct way to satisfy the four criteria to use flight data or flight-like test data. For an ideal fit the hardware on heritage missions must have the same functional performance requirements, the same operating and environmental conditions, and the same manufacturing process (design, fabrication, assembly, integration, and test) as the hardware that will be used on the new mission. Assuring a high degree of similitude between new and heritage hardware with respect to these characteristics will implicitly satisfy the criterion that no potentially significant risk contributor for the hardware is omitted from the PRA.

When comparing the manufacturing processes between heritage and new hardware, the following attributes must be considered:

1. **Manufacturer** - are the differences in manufacturers (if any) significant with respect to hardware reliability?
2. **Hardware Quality** - are there any differences in the quality control processes used by the manufacturers? The more rigorous the quality control process applied, the higher confidence there should be that the hardware will perform in a reliable manner.
3. **Hardware Design Process** – are there differences in the procedure was used to design the hardware? For example, JPL Design Principles tend to result in conservative design requirements. However, individual projects may waive these principles (if they obtain the appropriate approvals) and utilize a less conservative design process.

4. **Hardware Qualification/Test Program** - are the differences between past and future missions are significant enough to change the expected reliability of hardware between the two missions? Some missions may subject hardware to rigorous testing, while the same hardware flown on another mission may have certain tests waived. Similarly, the number of tests performed to qualify hardware can vary, depending upon the degree of confidence specified by the design requirements.

When comparing the hardware utilization of between heritage and new missions, the following attributes must be considered:

1. **Functional** – are differences in the functional behavior of the hardware in the two missions significant enough to change the expected reliability of the hardware between the two missions?
2. **Operational** – are differences in the operating environment significant enough to change the expected reliability of hardware between the two missions? Evaluating how different flight environments impact hardware reliability is challenging if only elicitation is available. Preferably, the impact of different ambient environments on risk should be assessed using a physics-of-failure approach. Even if all other criteria are satisfied, flight or test data for hardware flown in an ambient environment having a potentially significant and different risk impact on the hardware than the mission being addressed in the PRA should not be used as heritage for the PRA.

More details on assessing hardware heritage are described in Ref. 2.

A.2 Determining the Probability of Applicability

This section describes the process by which expert evaluations are elicited and processed to determine the provenance of heritage reliability data. First, experts will be asked to provide an estimate of the probability that reliability evidence from candidate heritage missions is applicable to a hardware item being flown on a future mission. The experts estimate this probability by considering the attributes discussed in Section A.1, as illustrated in Figure A-1, and the relative weight of each attributes. These questions are intended to help the expert ascertain the degree to which a potential heritage mission is applicable to the future mission for which data are needed.

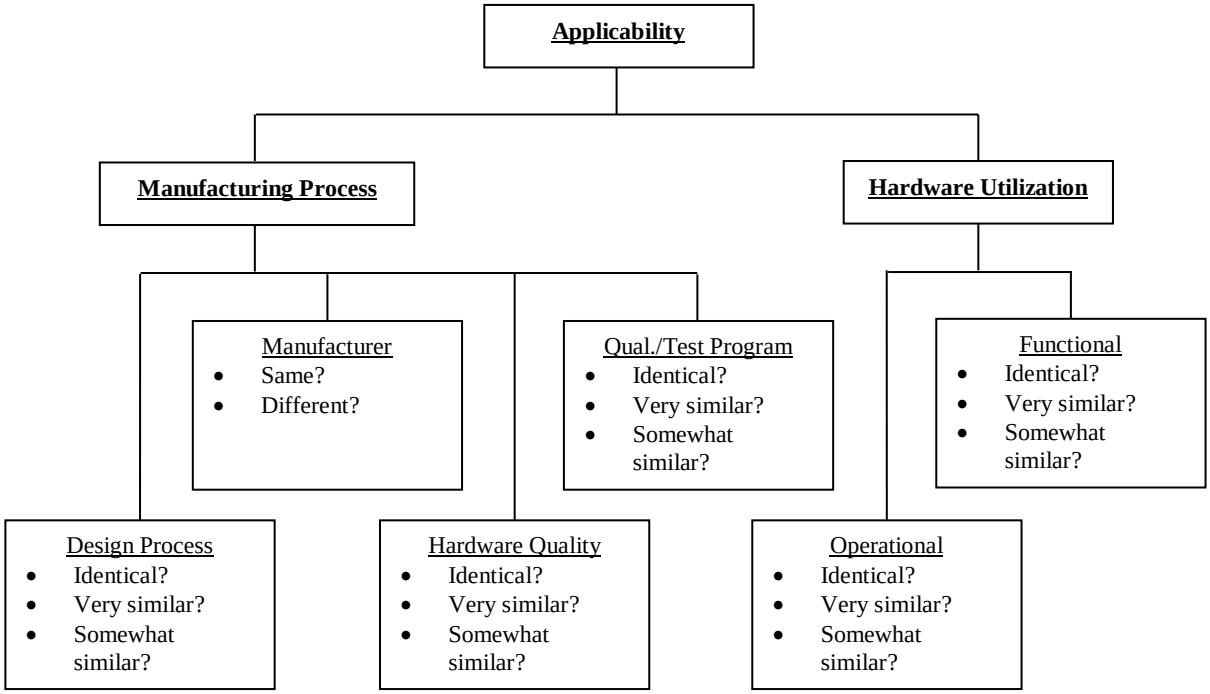

Figure A-1 Preliminary Applicability Tree

The questions listed in Figure A-1 are not meant to be mutually exclusive or exhaustive. Experts also may provide other attributes that are relevant to a particular case, and the relative importance of each attribute. The elicitation team then processes the expert elicitations using the seven step evaluation below:

1. For missions omitted by some experts, contact the other experts and ask for their evaluations of these missions. If the other experts argue that these missions are irrelevant, this means they apply a probability of zero.
2. Do a preliminary evaluation of the expert input. The objective of this evaluation is to increase the chances of developing a consensus estimate for each component on each individual mission identified by the experts. When completed (in Steps 4 and 5), the result will be a probability, for each individual mission and component, that the evidence from that mission is relevant to the component when it is flown on the future mission.
3. When evaluating the expert input, see whether the expert estimates indicate substantive agreement or disagreement. Have the experts supplied sufficient information for you to understand why they agree or disagree? Do the experts appear to be confident or uncertain in their assessments? Is it evident why? The worksheets in the Heritage Spreadsheet will be especially helpful in supporting this effort.
4. Proceed to the deliberation step of the process in which the experts are given the opportunity to explain their estimates to you and the other experts. The deliberation may be conducted either in a meeting or in a conference call. Past experience with such deliberations suggests that apparent differences in expert estimates are frequently due to different interpretations of aspects of the issue. The intent of the deliberation is to achieve a common understanding of the issue.
5. After the experts appear to have a common understanding of the issue, proceed with the evaluation of a consensus relevance probability. This probability does not have to be an analytical result from their original estimates such as the mean value. The deliberation may lead the experts to a consensus value although you may have to suggest an initial estimate based on what you have concluded in Step 2 and heard at the meeting.
6. If the experts are unable to reach consensus and persist in their different evaluations, do not attempt to force a consensus value. Bring the disagreement to the attention of the PRA lead (i.e., the Project Systems Engineer or designee). Be prepared to provide your evaluation of the expert arguments and to state what probability you believe should be assigned to the applicability of that mission and why. Your probability does not necessarily have to be the probability of one of the experts. Seek a resolution with the PRA lead, possibly involving additional experts.
7. The final part of the elicitation is estimating the probabilities that different combinations of heritage evidence are relevant to the future mission. Guidance for this step is furnished in Section A.3
8. Document the elicitation results so that they can be readily incorporated into the project PRA.

The result of this evaluation will be a set of weighted evidence sets which, when combined with the prior probability density function for the hardware item will furnish a set of weighted probability density functions. The sum of these functions provides an estimate of epistemic uncertainty for the failure parameter being considered. With respect to this estimate, the weights result from the elicitation.

The results from the expert elicitation and elicitation team should be stored in the system model along with the selected reliability value. This provides a rich justification for the selected reliability values within the system model.

A.3 Heritage Mission Combinations

First consider two potentially relevant missions, X and Y. Let their respective probabilities of relevance be Pr(X) and Pr(Y). These probabilities will result from the elicitation process.

There are four possible combinations of evidence associated with these two missions (a bar over a letter denotes the complement of the set represented by the letter):

1. both missions are relevant, $\left(\mathrm{X} \cap \mathrm{Y}\right)$;
2. only mission X is relevant, $\left(\mathrm{X} \cap \overline{\mathrm{Y}}\right)$;
3. only mission Y is relevant, $\left(\overline{\mathrm{X}} \cap \mathrm{Y}\right)$; and
4. neither mission is relevant, $\left(\overline{\mathrm{X}} \cap \overline{\mathrm{Y}}\right)$;

The issue is how to quantify the probabilities evidence from these potentially relevant missions apply to the future mission, for the various combinations listed above. Specifically, it is necessary to quantify the expressions:

- $\Pr(X \cap Y)$;
- $\Pr(X \cap \overline{Y})$;
- $\Pr(\overline{X} \cap Y)$; and
- $\Pr(\overline{X} \cap \overline{Y})$.

In evaluating these probabilities, we have to consider the relevance of X to Y. Suppose that missions X and Y are totally relevant to each other. That is, hypothesize that every attribute of the component flown on mission X is identical to the attribute on mission Y. Then:

$$X = Y \tag{A-1}$$
$$\Pr(X) = \Pr(Y) \tag{A-2}$$

$$\Pr(X \cap Y) = \Pr(X) = \Pr(Y) \tag{A-3}$$
$$\Pr(X \cap \overline{Y}) = \Pr(\overline{X} \cap Y) = 0 \tag{A-4}$$

and:

$$\Pr(\overline{X} \cap \overline{Y}) = 1 - \Pr(Y) = 1 - \Pr(X) \tag{A-5}$$

An alternative example involves the experts deciding that relevance among different missions is essentially dependent on just two attributes, the:

1. process employed to design the component; and
2. qualification/test program to which the component was submitted.

The consensus, in this example, is that negligible importance should be assigned to all other attributes.

Imagine, for the purpose of this alternative example, that according to responses in the Heritage Spreadsheet:

- both attributes have comparable importance with respect to determining relevance to the future mission;
- the process employed to design the component flown on mission X is identical to the future mission, but the qualification/test programs are highly dissimilar; and
- the process employed to design the component flown on mission Y is very different from the process being applied to the future mission, but the qualification/test programs are equivalent with respect to relevance.

In this second example, it is unclear how to interpret the intersection of X and Y. Section A.2 explicitly states that the Figure A-1 attributes are not meant to be mutually exclusive, and it would be inordinately restrictive to assume that the intersection of X and Y is null. A recommended first approximation in such situations is to model the probabilities as independent. Thus:

$$\Pr(X \cap Y) = \Pr(X)\Pr(Y) \tag{A-6}$$
$$\Pr(X \cap \overline{Y}) = \Pr(X)[1 - \Pr(Y)] \tag{A-7}$$
$$\Pr(X \cap \overline{Y}) = \Pr(Y)[1 - \Pr(X)] \tag{A-8}$$

and:

$$\Pr(\overline{X} \cap \overline{Y}) = [1 - \Pr(Y)][1 - \Pr(X)] \tag{A-9}$$

The basic proposal, then, is that if two missions have comparable attributes (i.e., there is strong relevance between them), then Eqs. A-3 through A-5 should be used to estimate joint probabilities of their relevance to the future mission. If two missions have dissimilar attributes (such as in the second example), Eqs. A-6 through A-9 should be applied.

The basic proposal requires the analysts to infer information from the Heritage Spreadsheet and, if appropriate, supplemental discussions with the experts. It is essential, therefore, that the quantified inferences be vetted. In order to accomplish this it will be necessary for the analysts to present their results to the experts, along with the rationale for their estimates. If the analysts and experts can achieve a consensus, then there will be appreciable

confidence in the relevance probabilities. If a consensus cannot be achieved the analysts should proceed as in Step 5 of Section A.2.

During discussions between the analysts and experts, each has a specific role. The analysts must ensure compliance with the mathematics of probability. This should be accomplished from the perspective of helping the experts understand how their opinions regarding relevance from one mission may relate to the relevance of evidence from another mission (i.e., the analysts facilitate expert calibration). Instead of merely comparing individual missions to Juno, the experts must now consider similarities among the various heritage missions.

The role of the experts is to furnish the probabilities. However, attempting to elicit detailed probabilities for the various set intersections is not recommended. Typically, the experts will not be totally comfortable quantifying such relationships and the quality of the results may be diminished as a consequence. The challenge to the elicitors is to obtain as much information from the experts as the latter can comfortably provide, without overtaxing or frustrating them with excessive details.

The ultimate objective of the elicitation is to obtain a posterior probability density function, $\pi_1(\phi|E)$, for the heritage evidence. Here the variable, ϕ, can be either a failure rate (e.g., per hour) or probability of failure (e.g., per demand). If $\pi_0(\phi)$ symbolizes the prior probability density function, then:

$$\pi_1(\phi|E) = \Pr(X \cap Y)\,\pi_1\left[\phi\middle|(X \cap Y)\right] + \Pr(X \cap \overline{Y})\,\pi_1\left[\phi\middle|(X \cap \overline{Y})\right] + \Pr(\overline{X} \cap Y)\,\pi_1\left[\phi\middle|(\overline{X} \cap Y)\right] + \Pr(\overline{X} \cap \overline{Y})\,\pi_0(\phi) \tag{A-10}$$

The evidence, given that both missions X and Y are relevant, is the sum of the failure counts for each mission and the aggregate exposure history for the two missions (i.e., the sum of the operating times for each mission, or the total number of demands over the missions).

Given that only mission X is relevant, the evidence is simply the failure count and exposure history for mission X. Similarly, if only mission Y is relevant the evidence is the failure count and exposure history from mission Y. If neither mission is relevant there is no heritage evidence, so we only have our prior knowledge.

Equation A-10 can readily be expanded to situations with more than two potentially relevant missions. If there are N potentially relevant missions, then there are N^2 combinations of evidence. This can become rather cumbersome numerically, but often several candidate heritage missions are so similar that the probability only some, but not all mission are relevant is negligible. In such instances individual missions can be grouped, which minimizes the effective size of N.

Acknowledgments

The work described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration, as well as funding from JPL burden funds.

References

[1]Deckert, George. "NASA Hazard Analysis Process." NASA Technical Report Server (NTRS) 20100040678, Nov. 29, 2010. https://ntrs.nasa.gov/search.jsp?R=20100040678

[2]Speece, D. J., "Objective Criteria for Heritage Hardware Reuse", Aerospace Report TOR-2010(8591)-19, June 30, 2010.

[3]Schreiner, S., Rozek, M. L., Kurum, A., Everline, C. J., Ingham, M. D., & Nunes, J. "Towards a methodology and tooling for Model-Based Probabilistic Risk Assessment (PRA)". *AIAA Space conference*, 2016, pp. 5545.